



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Trabalho Prático n.º 1

Compressão de Imagem

Multimédia

Licenciatura em Engenharia Informática

2021/2022

Duarte Emanuel Ramos Meneses – 2019216949 – duartemeneses@student.dei.uc.pt

Inês Martins Marçal – 2019215917 – inesmarcal@student.dei.uc.pt

Patrícia Beatriz Silva Costa – 2019213995 – patriciacosta@student.dei.uc.pt

Índice

Introdução.....	3
Pergunta 1.4 – Compressão com editor.....	4
Pergunta 5.4 – RGB vs YCbCr.....	4
Pergunta 6.3 - Downsampling.....	7
Pergunta 7.1.3 - DCT	8
Pergunta 7.2.3 – DCT em blocos 8x8.....	10
Pergunta 7.3.2 - DCT em blocos 64x64	12
Pergunta 8.3 – Quantização	14
Pergunta 8.4 – Quantização vs DCT	22
Pergunta 9.3 - DPCM	22
Pergunta 10.2 – Erros e imagens descomprimidas.....	29
Conclusão	37
Referências.....	38

Introdução

Atualmente, com a quantidade de informação gerada no mundo, é cada vez mais necessário comprimir essa informação, uma vez que os recursos de armazenamento são limitados. Este é claramente um ponto fulcral da unidade curricular de Multimédia.

Tendo esta questão como ponto de partida, este trabalho prático tem o intuito de explorar mais detalhadamente os passos do *codec* de compressão de imagem JPEG, recorrendo à linguagem de programação Python.

Ao longo deste relatório vamos apresentando os resultados de cada passo do *codec*, bem como explicações para esses mesmos resultados. Para obter esses resultados, utilizamos três imagens distintas entre si para visualizarmos bem em que tipo de imagens este *codec* atua melhor.

Pergunta 1.4 – Compressão com editor

Comprimindo as imagens bmp no formato jpeg utilizando um editor de imagem (GIMP), obtivemos os seguintes resultados:

Qualidade	Tamanho em disco		
	barn_mountains.bmp	logo.bmp	peppers.bmp
Original	349 KB	412 KB	577 KB
Alta (75%)	34 KB	10 KB	31 KB
Média (50%)	22 KB	8 KB	21 KB
Baixa (25%)	14 KB	7 KB	14 KB

Como podemos verificar analisando os dados da tabela acima, quando se comprime uma imagem, o tamanho em disco da resultante é inferior ao original (qualquer que seja a qualidade dessa compressão). Fica ainda evidente nos dados acima que quanto mais baixa a qualidade de compressão, menor o espaço de armazenamento necessário.

Estas compressões foram feitas, aproximadamente, nas seguintes proporções:

Qualidade	Proporção de compressão		
	barn_mountains.bmp	logo.bmp	peppers.bmp
Alta (75%)	10:1	41:1	19:1
Média (50%)	16:1	52:1	27:1
Baixa (25%)	25:1	59:1	41:1

Cada proporção ajuda-nos a entender melhor a quantidade de compressão existente. Por exemplo, se a proporção for de 10:1 significa que 10 pixéis se tornaram em apenas 1. Deste modo, fica evidente que quanto maior for o valor do lado esquerdo da proporção, maior a taxa de compressão.

Quanto à qualidade das imagens resultantes, tal como era esperado, quanto menor a qualidade da compressão, menor a qualidade do resultado final. Quanto pior qualidade de compressão se tem, maior a perda de informação, logo mais degradada a imagem ficará ao olho humano. Por exemplo, qualquer que seja a imagem, ao diminuir a qualidade de compressão, a resultante fica com grão.

Na imagem do logo, qualquer tipo de compressão causa ruído. Isto acontece, pois, o JPEG é bom para imagens com transições suaves. Para os logos, a melhor solução é o PNG.

Pergunta 5.4 – RGB vs YCbCr

Como no modelo RGB existe redundância da luminância nos 3 canais (R, G e B), o modelo YCbCr tem como intuito reduzir essa mesma redundância. Deste modo, o YCbCr separa a luminância da componente cromática. Esta última é a menos sensível ao olho humano.

Aplicando um colormap para tons de cinzento a cada canal do modelo YCbCr, obtemos os seguintes resultados para a imagem peppers.bmp:

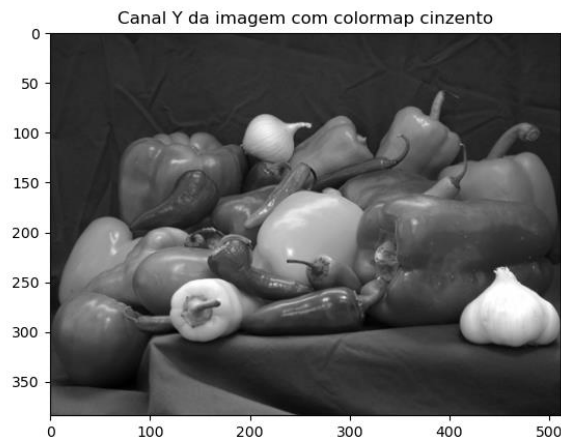


Figura 1 - Canal Y do modelo YcbCr da imagem peppers.bmp utilizando um colormap cinzento

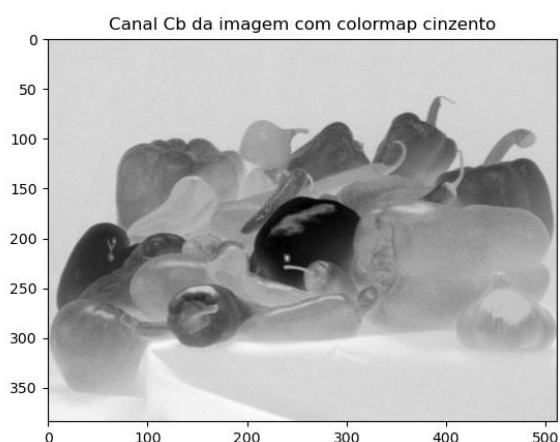


Figura 2 - Canal Cb do modelo YcbCr da imagem peppers.bmp utilizando um colormap cinzento

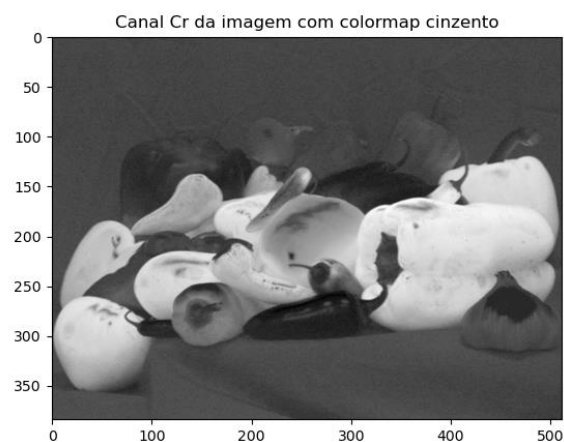


Figura 3 - Canal Cr do modelo YcbCr da imagem peppers.bmp utilizando um colormap cinzento

Analisando as imagens acima comprovamos o que enunciamos anteriormente. Efetivamente, é no canal Y que conseguimos visualizar a imagem com mais detalhe. Isto acontece uma vez que este canal é o que contém a luminância, enquanto os Cb e Cr contêm a cromaância. Como o olho humano é mais sensível à luminância (canal Y) que à cromaância (canais Cb e Cr), é a figura 1 que apresenta mais detalhe.

É fácil de perceber, portanto que, sendo os canais Cb e Cr os que menos detalhe apresentam (menos relevância na construção da imagem pois o olho humano não é tão sensível à cromaância, propriedade destes canais), são os que servirão de base à compressão. Menor informação nestes canais em pouco influenciará a percepção humana da imagem no seu todo. Já se essa compressão fosse realizada no canal Y, o olho humano detetaria mais facilmente.

Os resultados da imagem peppers.bmp dividida nos seus canais R, G e B, cada um com o colormap adequado (cada canal com um colormap da cor em questão) são os seguintes:

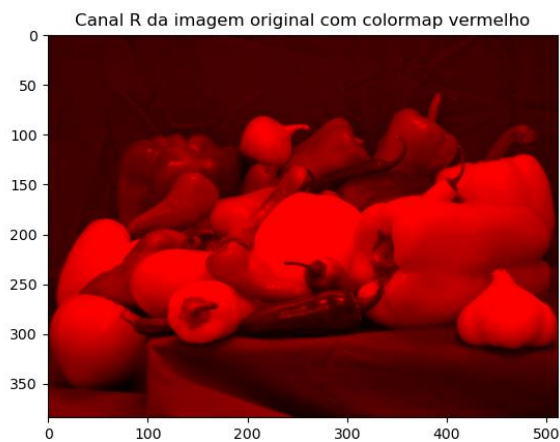


Figura 4 - Canal R do modelo RGB da imagem peppers.bmp utilizando um colormap vermelho

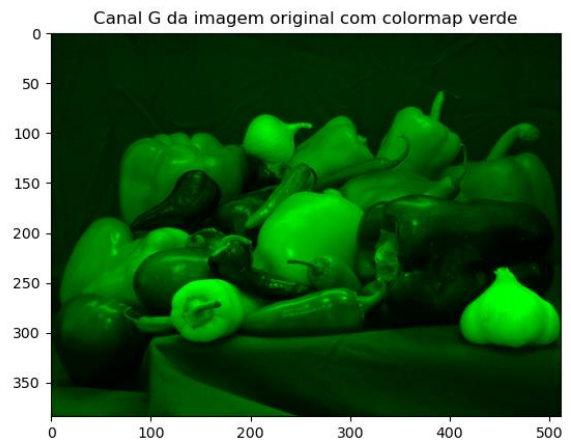


Figura 5 - Canal G do modelo RGB da imagem peppers.bmp utilizando um colormap verde

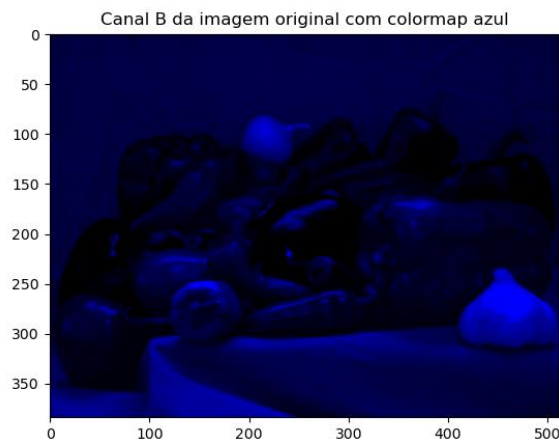


Figura 6 - Canal B do modelo RGB da imagem peppers.bmp utilizando um colormap azul

Comparando agora o canal Y do modelo YCbCr com os canais R, G e B do modelo RGB, fica evidente que existe detalhe em todos. Isto acontece, pois, tal como já referimos em cima, no modelo RGB existe redundância da luminância nos 3 canais. Sendo esta a que está presente no canal Y do YCbCr, é normal que o nível de detalhe seja similar aos canais R, G e B.

Uma questão que nos fez refletir foi o facto de o nível de detalhe entre os canais R, G e B serem ligeiramente diferentes. O canal G é o que apresenta mais detalhe, seguido do R e, por último, o B. Isto acontece, pois, o olho humano é mais sensível às tonalidades verdes, depois às vermelhas e só no fim às azuis. Desta forma, apesar de ambas apresentarem luminância (propriedade a que o olho humano é mais sensível), a cor verde é a que consegue representar mais fielmente os detalhes da imagem original.

Pergunta 6.3 - Downsampling

No downsampling, o número de pixels de uma imagem diminui. Se a proporção for 4:2:2, a imagem resultante terá metade das colunas. Se for 4:2:0, terá metade das colunas e metade das linhas. Deste modo, fica evidente que com esta técnica o tamanho da imagem diminui, uma vez que se perde informação. Essa informação nunca mais é recuperada a 100%. Habitualmente utilizam-se duas técnicas: replicar a coluna/linha anterior ou fazer a média da anterior com a seguinte. Neste trabalho prático optamos por utilizar a técnica da replicação.

Os tamanhos resultantes da aplicação de downsampling são os seguintes:

- barn_mountains.bmp

Canal		Tamanho
Y	Original	(304, 400)
	Com downsampling 4:2:0	(304, 400)
	Com downsampling 4:2:2	(304, 200)
Cb	Original	(304, 400)
	Com downsampling 4:2:0	(152, 200)
	Com downsampling 4:2:2	(304, 200)
Cr	Original	(304, 400)
	Com downsampling 4:2:0	(152, 200)
	Com downsampling 4:2:2	(304, 200)

- logo.bmp

Canal		Tamanho
Y	Original	(288, 512)
	Com downsampling 4:2:0	(288, 512)
	Com downsampling 4:2:2	(288, 512)
Cb	Original	(288, 512)
	Com downsampling 4:2:0	(144, 256)
	Com downsampling 4:2:2	(288, 256)
Cr	Original	(288, 512)
	Com downsampling 4:2:0	(144, 256)
	Com downsampling 4:2:2	(288, 256)

- peppers.bmp

Canal		Tamanho
Y	Original	(384, 512)
	Com downsampling 4:2:0	(384, 512)
	Com downsampling 4:2:2	(384, 512)
Cb	Original	(384, 512)
	Com downsampling 4:2:0	(192, 256)
	Com downsampling 4:2:2	(384, 256)
Cr	Original	(384, 512)
	Com downsampling 4:2:0	(192, 256)
	Com downsampling 4:2:2	(384, 256)

Tal como era espectável, o canal Y não sofreu alterações em nenhuma imagem, quer no número de colunas quer no número de linhas nas duas variantes de downsampling. Isto acontece, pois, a proporção é de 4:2:2 ou 4:2:0, o que indica que os canais Cb e Cr tem a sua taxa de amostragem reduzida para metade na direção horizontal (no caso de 4:2:2) ou que estes mesmo dois canais têm a sua taxa de amostragem reduzida para metade em ambas as direções (no caso de 4:2:0).

Os canais Cb e Cr são os que sofrem compressão uma vez que são os que o olho humano é menos sensível. O Y, sendo o que é mais sensível, não sofre compressão para a imagem não ficar distorcida ao olho humano.

Num exemplo mais prático, tendo uma matriz 10x10, com downsampling 4:2:2, temos uma taxa de compressão de 100:50 que é equivalente a 2:1.

Já com downsampling 4:2:0, obtemos uma taxa de compressão de 100:25 que é equivalente a 4:1.

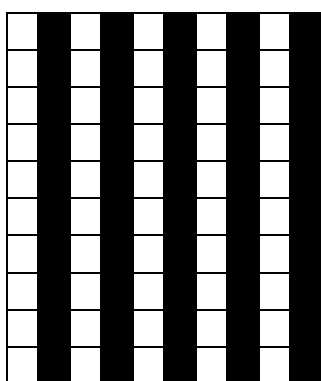


Figura 7 - Representação de downsampling 4:2:2 em matriz 10x10

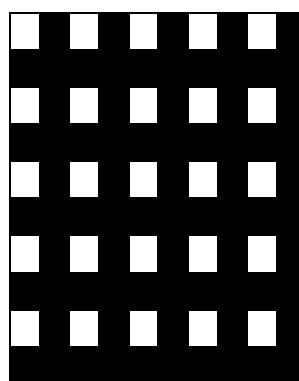


Figura 8 - Representação de downsampling 4:2:0 em matriz 10x10

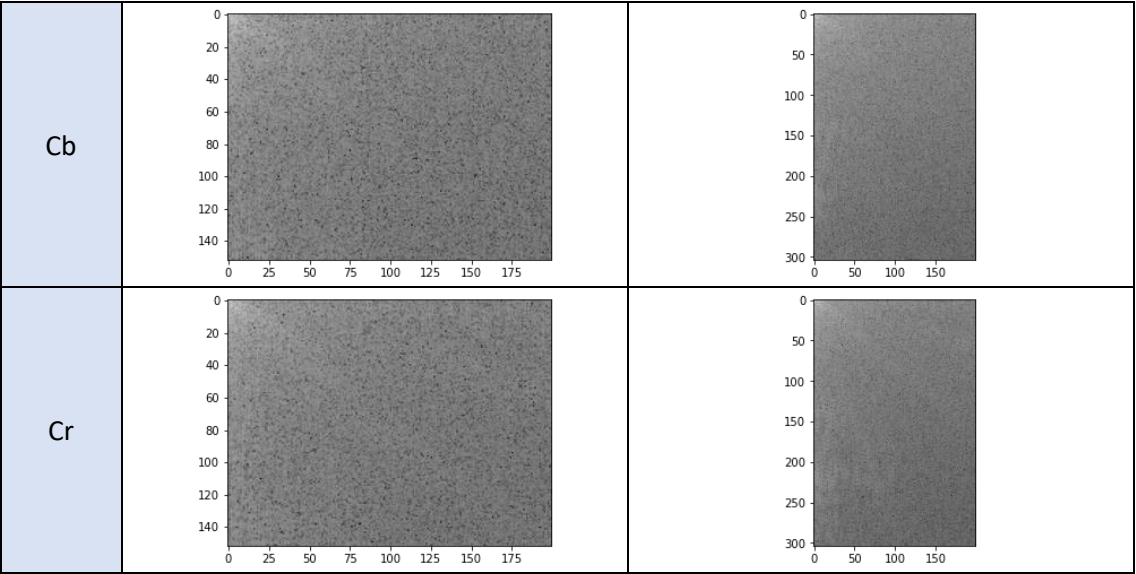
Fica assim evidente que a proporção 4:4:2, reduzindo apenas a taxa de amostragem na direção horizontal, tem menor destrutividade que a proporção 4:4:0 que reduz a taxa de amostragem em ambas as direções.

Pergunta 7.1.3 - DCT

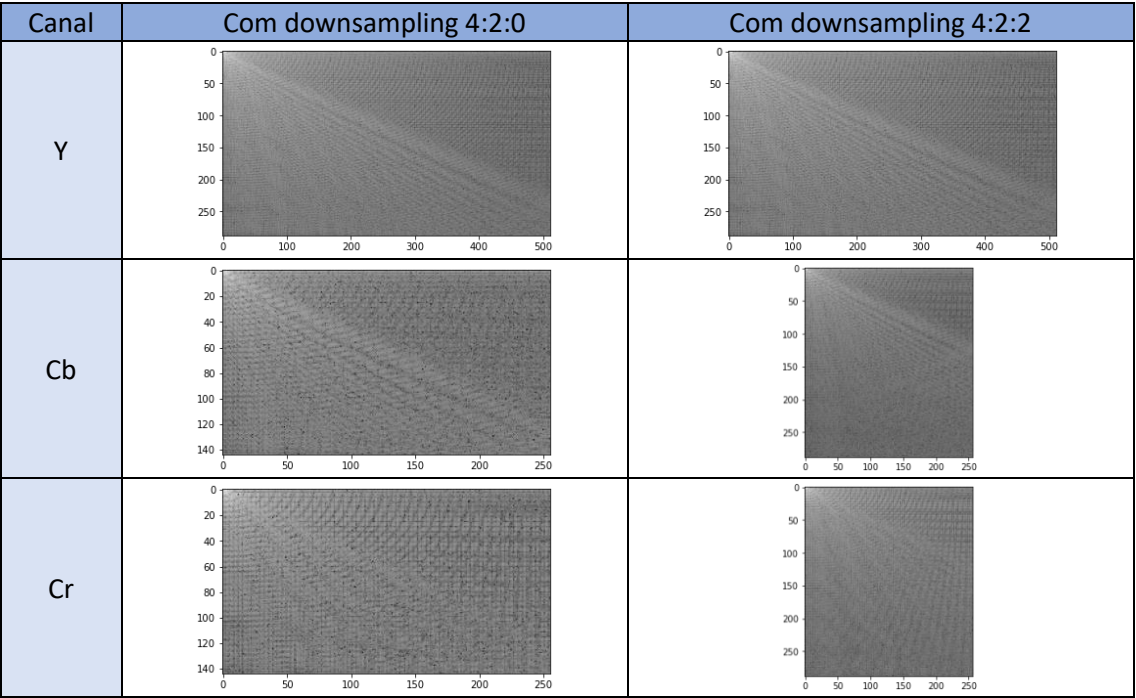
A DCT permite converter a imagem para o domínio de frequência, o que leva a que as baixas frequências espaciais se situem no canto superior esquerdo da matriz e as elevadas no canto inferior direito. Isto permite que exista mais compressão uma vez que a maioria da energia se situa concentrada em determinados pixéis.

- barn_mountains.bmp

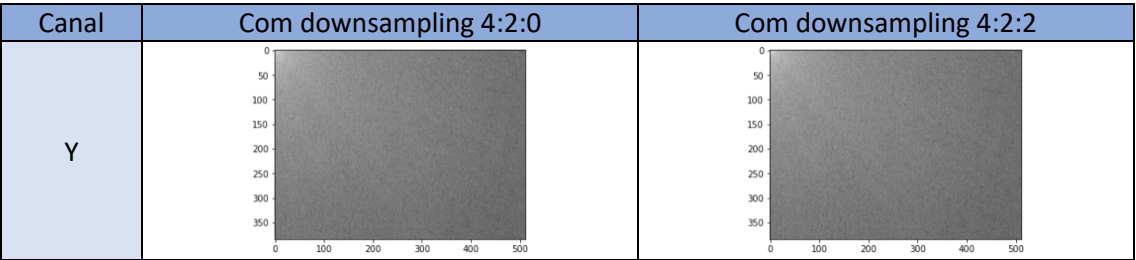
Canal	Com downsampling 4:2:0	Com downsampling 4:2:2
Y		

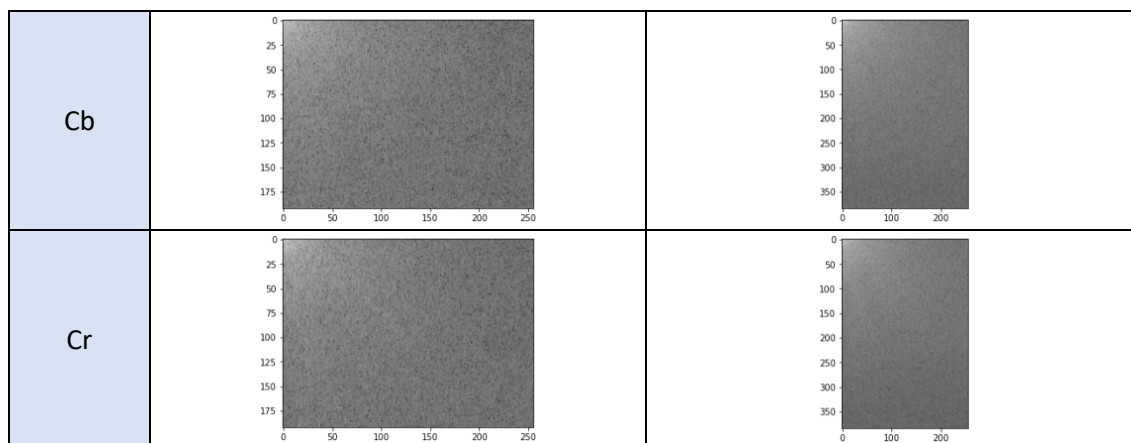


- logo.bmp



- peppers.bmp





Como podemos visualizar pelas imagens acima, percebemos que uma concentração da maior parte das frequências da imagem original num número reduzido de pixéis. Isto vai, naturalmente, ajudar à compressão.

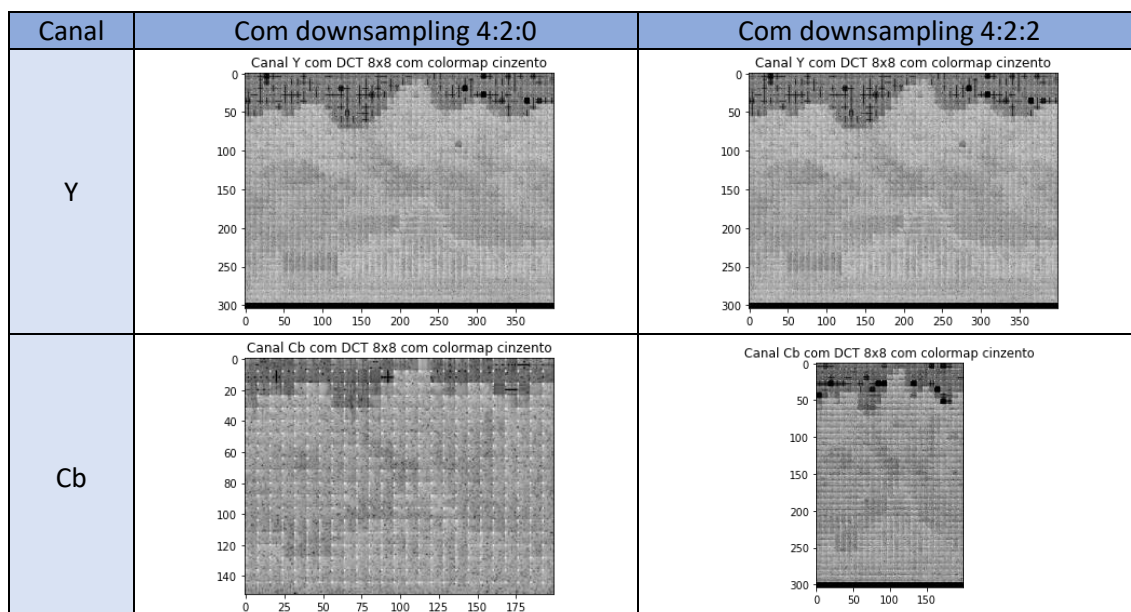
Posto isto, pelas propriedades de compactação de energia, a DCT apresenta melhores resultados em imagens suaves, uma vez que maior parte da energia se encontra nas baixas frequências espaciais, o que permite que poucas células de frequência contendam a maior parte da energia da imagem.

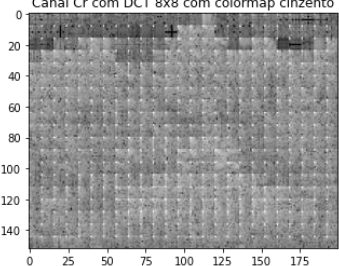
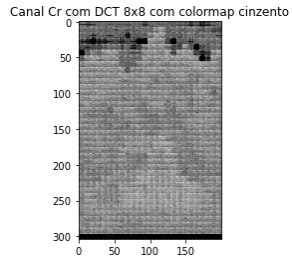
É também evidente que as imagens não são distinguíveis pela sua representação após aplicar a DCT.

Pergunta 7.2.3 – DCT em blocos 8x8

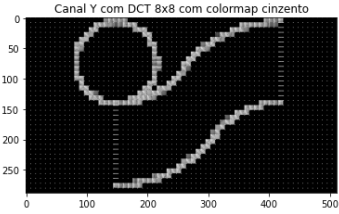
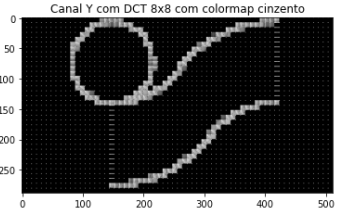
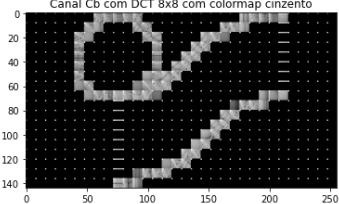
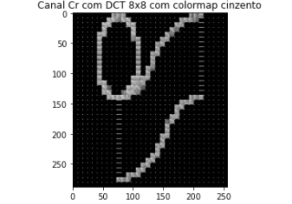
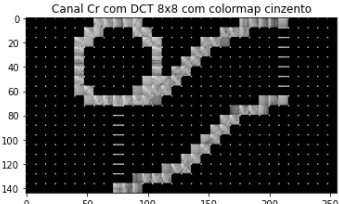
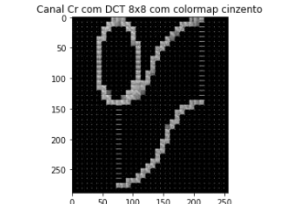
Neste caso em que aplicamos a DCT em blocos, reduzimos a probabilidade da ocorrência de transições abruptas de frequência, o que favorece a utilização da DCT. Podemos comprovar isso nos seguintes resultados:

- barn_mountains.bmp

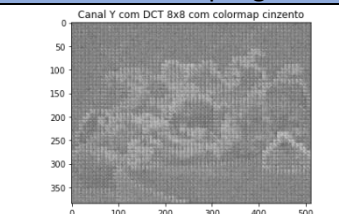
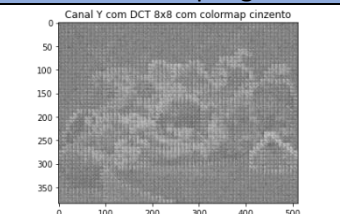
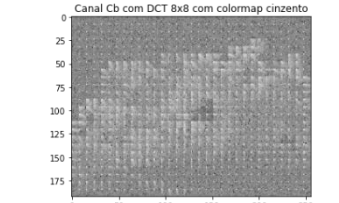
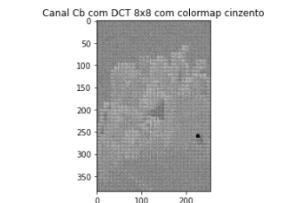


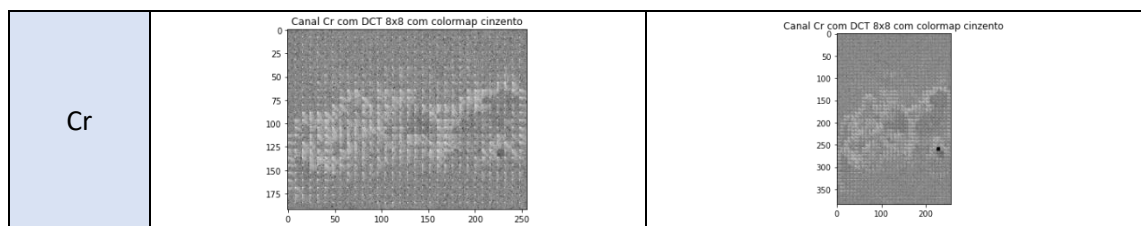
Cr	<p>Canal Cr com DCT 8x8 com colormap cinzento</p> 	<p>Canal Cr com DCT 8x8 com colormap cinzento</p> 

- logo.bmp

Canal	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	<p>Canal Y com DCT 8x8 com colormap cinzento</p> 	<p>Canal Y com DCT 8x8 com colormap cinzento</p> 
Cb	<p>Canal Cb com DCT 8x8 com colormap cinzento</p> 	<p>Canal Cr com DCT 8x8 com colormap cinzento</p> 
Cr	<p>Canal Cr com DCT 8x8 com colormap cinzento</p> 	<p>Canal Cr com DCT 8x8 com colormap cinzento</p> 

- peppers.bmp

Canal	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	<p>Canal Y com DCT 8x8 com colormap cinzento</p> 	<p>Canal Y com DCT 8x8 com colormap cinzento</p> 
Cb	<p>Canal Cb com DCT 8x8 com colormap cinzento</p> 	<p>Canal Cb com DCT 8x8 com colormap cinzento</p> 



Comparando estes resultados com os anteriores fica claro que aplicando a DCT em blocos 8x8, a imagem resultante fica mais semelhante à original, o que permite distingui-las entre si.

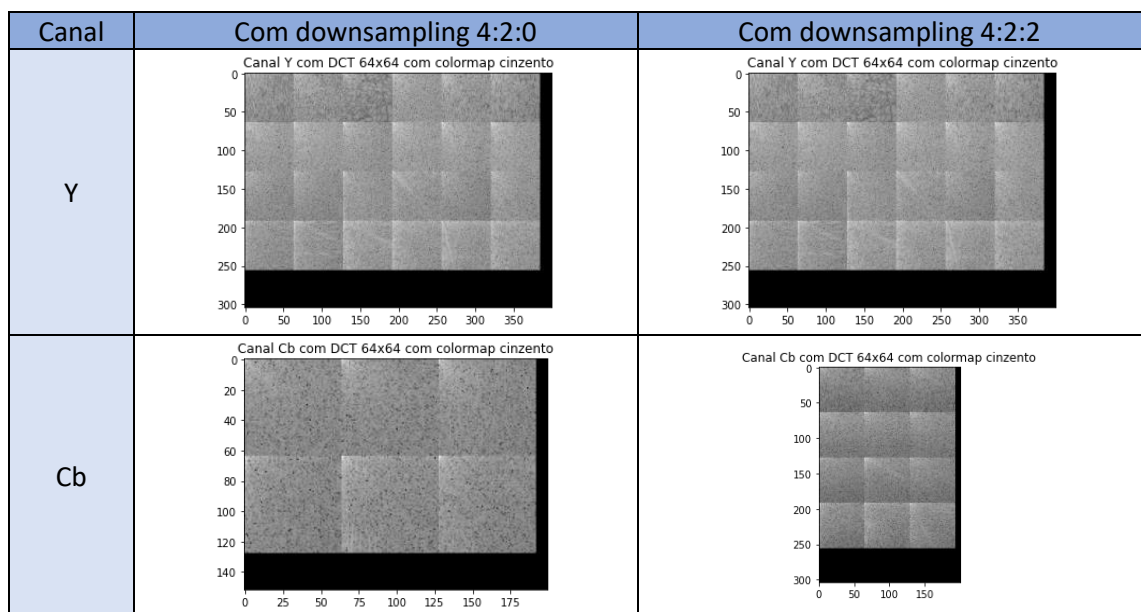
Tal como dissemos acima, a DCT tem melhor resultados em imagens suaves. Deste modo, analisando a imagem em blocos 8x8, reduzimos a probabilidade da ocorrência de transições abruptas de frequência, o que favorece a utilização da DCT em blocos ao invés de aplicar na imagem inteira. Como nos blocos reduzimos a probabilidade de ocorrência de transições abruptas, encontramos mais redundância de frequências, o que ajuda a compressão.

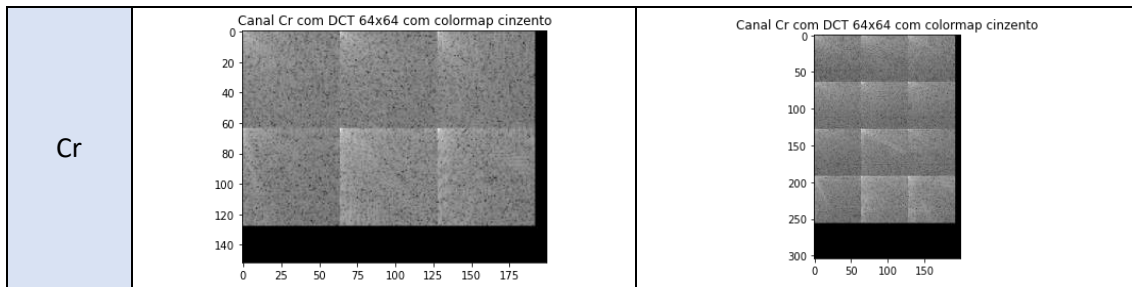
Posto isto, percebemos que aplicar a DCT em blocos (8x8 neste caso) favorece a compressão em relação ao modo ilustrado na alínea anterior.

Pergunta 7.3.2 - DCT em blocos 64x64

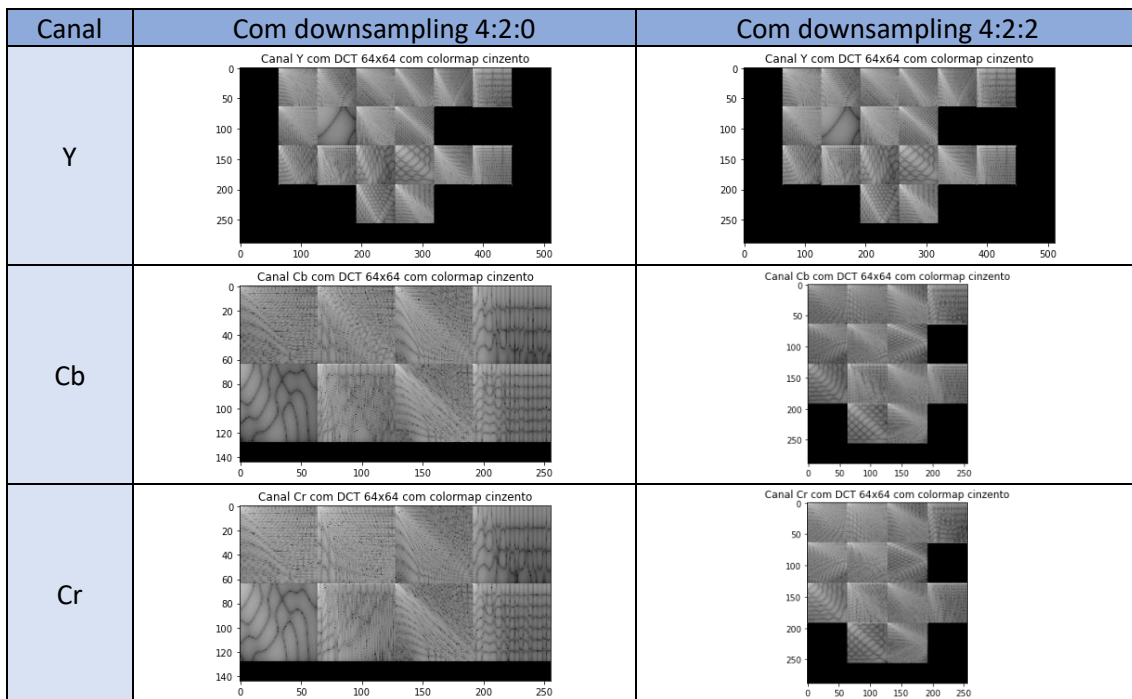
Uma vez que quanto menor o tamanho, mais se filtram as altas frequências, é espectável que aplicando DCT em blocos 64x64, se obtenham melhores resultados comparando a uma aplicação à imagem inteira. No entanto, espera-se que os resultados sejam piores do que os anteriores em que aplicamos em blocos 8x8.

- barn_mountains.bmp

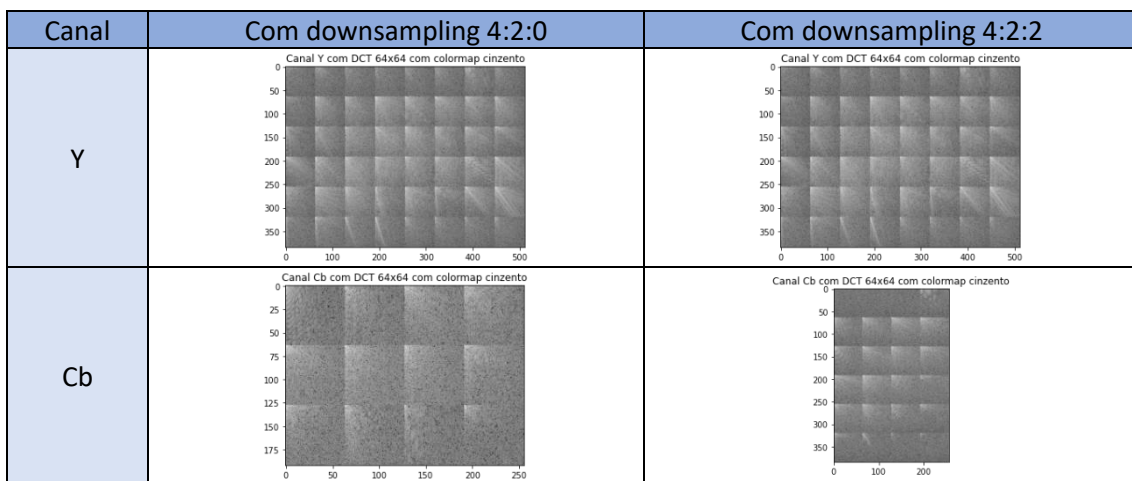


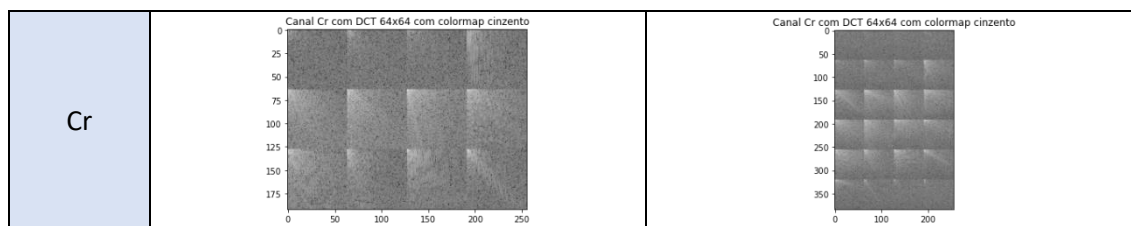


- logo.bmp



- peppers.bmp





Analisando as imagens anteriores, vemos que aqui volta a acontecer o que aconteceu na pergunta 7.1. As imagens não são distinguíveis pela DCT aplicada em blocos 64x64.

Isto acontece uma vez que a energia ao se espalhar pela zona correspondente (alta frequência no canto inferior direito e baixa frequência no canto superior esquerdo), vai desvirtuar a imagem. Mesmo fazendo em blocos, estes não têm um tamanho suficientemente pequeno para que, mesmo a energia estando espalhada, se perceba que imagem é. Esta situação tem como consequência a compressão não ser tão favorecida uma vez que existem transições mais abruptas de energia comparando com a alínea anterior.

No entanto, é melhor do que aplicar à imagem inteira, uma vez que assim sempre se filtram as frequências mais altas nesses blocos.

Concluimos assim que a resolução da DCT é sempre proporcional à dimensão dos blocos em que se aplica. Quanto menor o tamanho, mais se filtram as altas frequências e melhor será a compressão resultante.

Pergunta 8.3 – Quantização

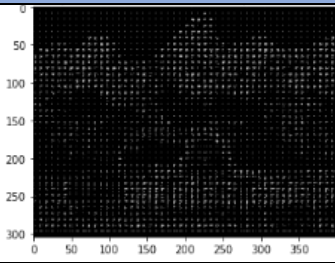
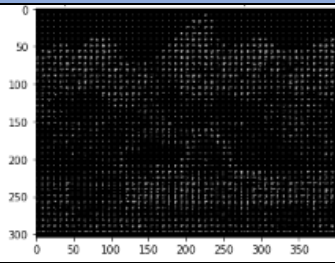
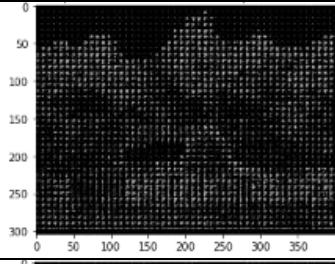
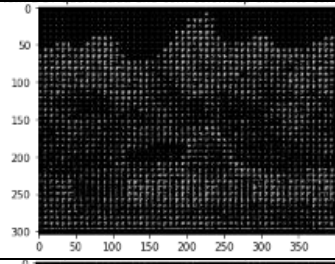
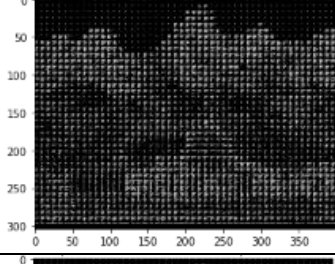
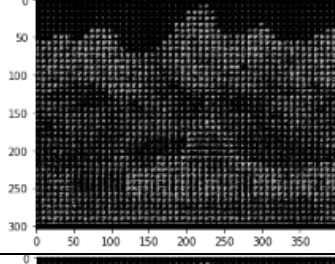
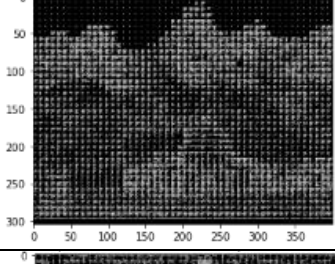
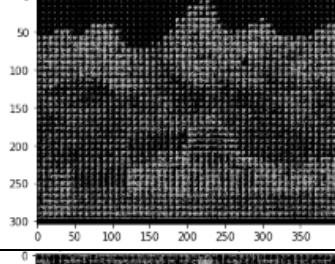
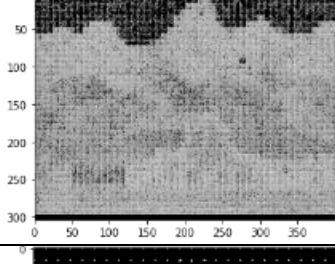
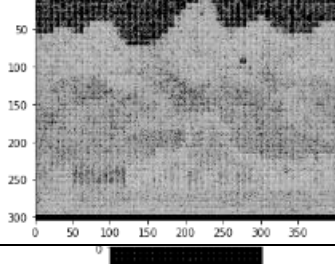
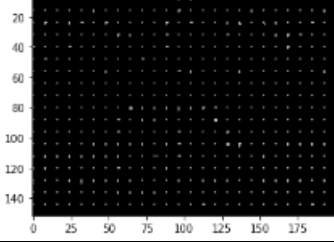
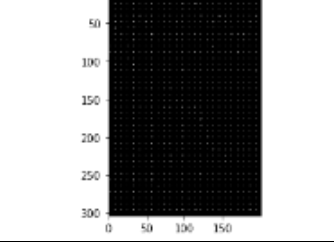
Aplicando uma matriz de quantização que provém de um dado fator de qualidade, conseguimos maximizar a taxa de compressão minimizando as perdas percetuais.

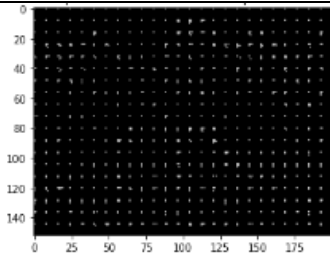
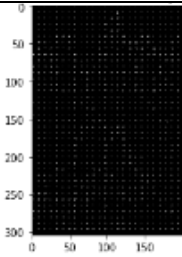
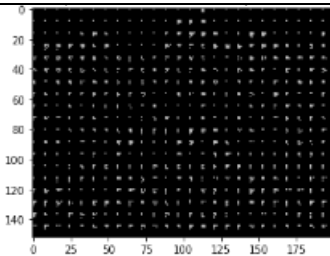
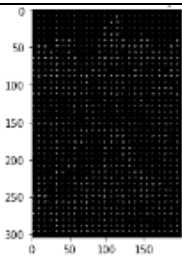
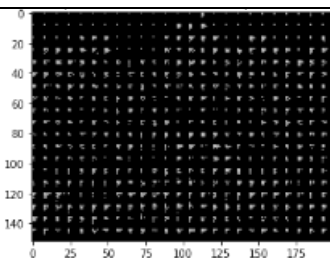
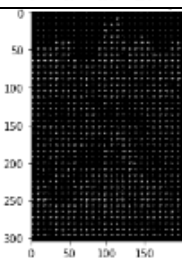
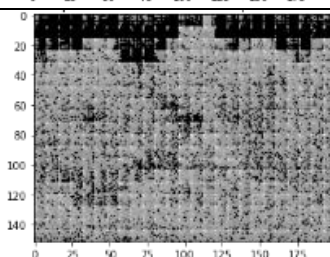
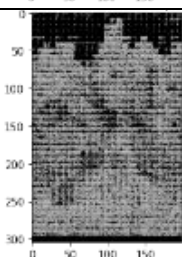
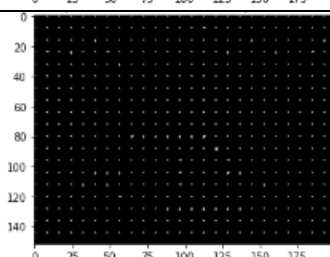
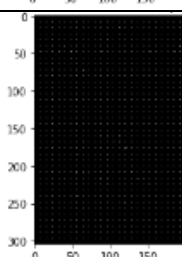
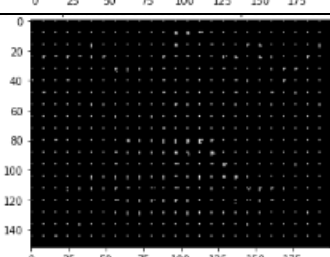
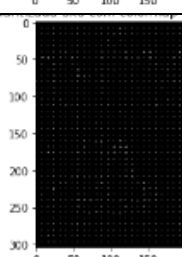
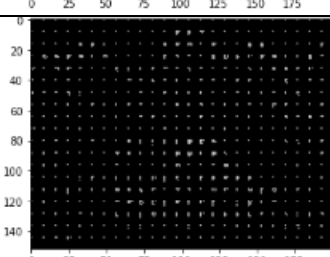
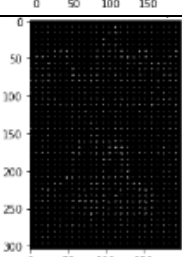
Quanto maior for esse fator de qualidade, menores serão os valores da matriz que vão dividir os valores da imagem, pelo que existirão menos frequências quase nulas. Com isto, obtemos menos informação perdida.

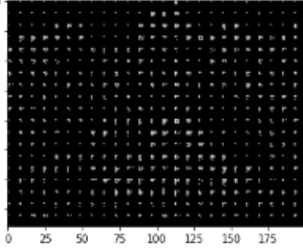
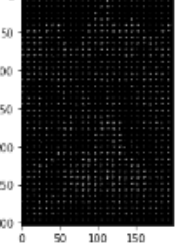
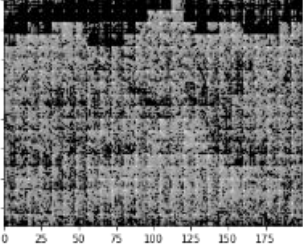
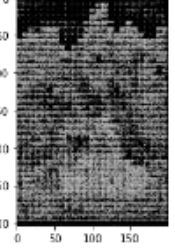
Com o objetivo de obter a matriz quantizada, é necessário, em primeiro lugar, multiplicar o fator de escala (obtido a partir do fator de qualidade) pela matriz de quantização. Posteriormente, ao dividirmos a DCT pela matriz resultante, iremos obter a matriz DCT quantizada. Se não existirem alterações em relação à DCT original, significa que não houve quantização pelo que a taxa de compressão será baixa.

Os resultados seguintes pretendem ilustrar o enunciado.

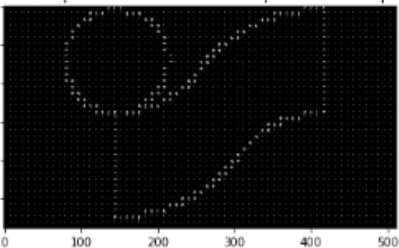
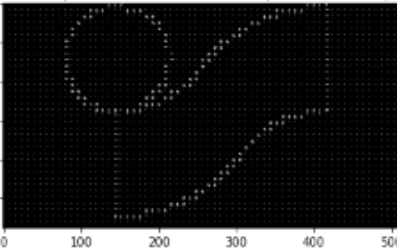
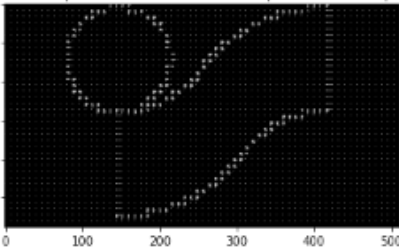
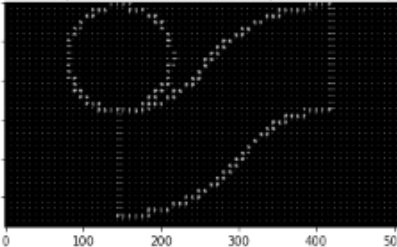
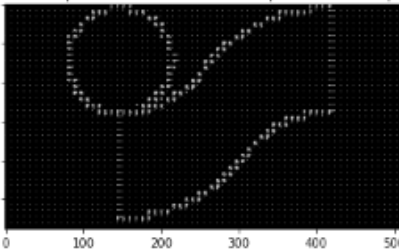
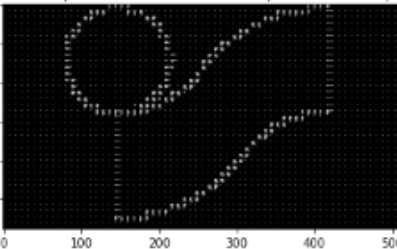
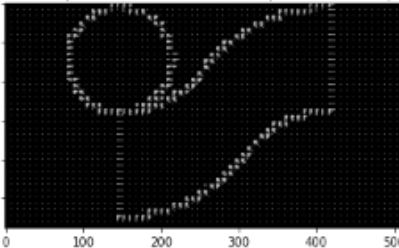
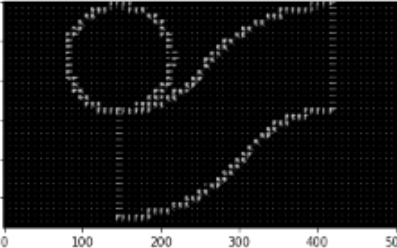
- barn_mountains.bmp

Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		
	25		
	50		
	75		
	100		
Cb	10		

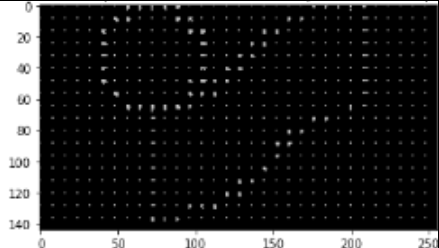
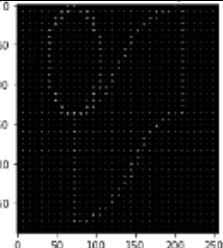
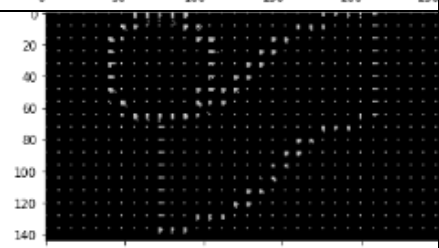
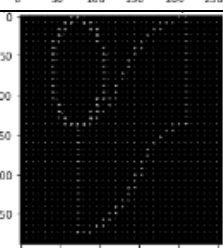
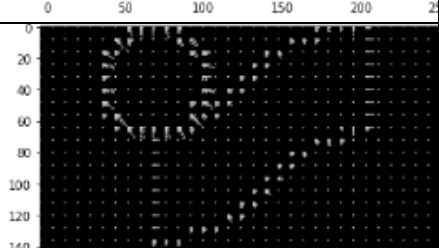
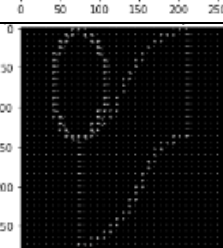
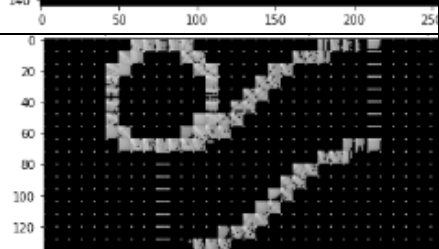
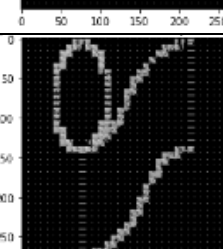
	25		
	50		
	75		
	100		
Cr	10		
	25		
	50		

	75		
	100		

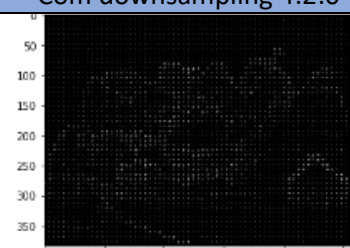

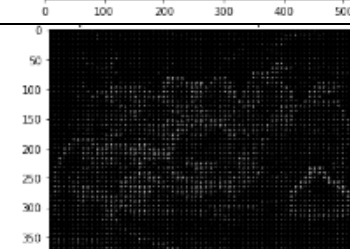
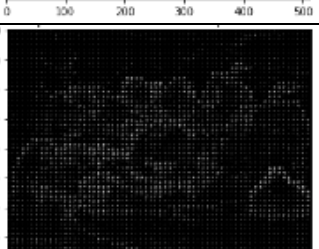
- logo.bmp

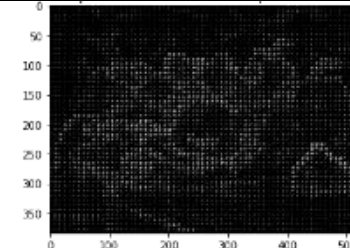
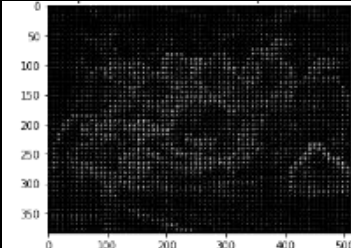
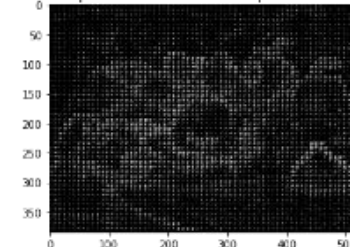
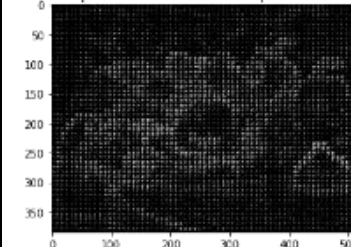
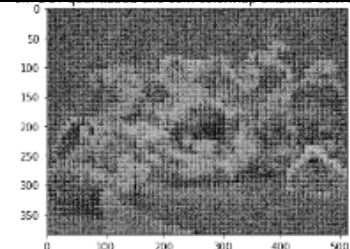
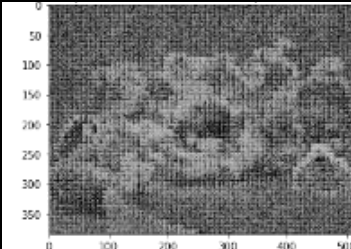
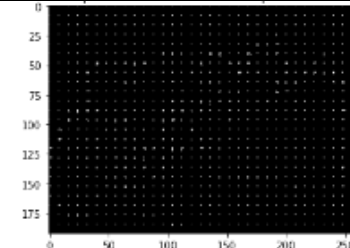
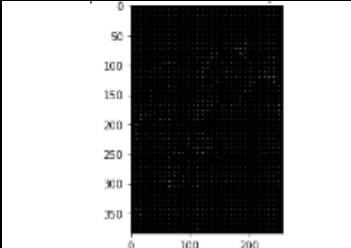
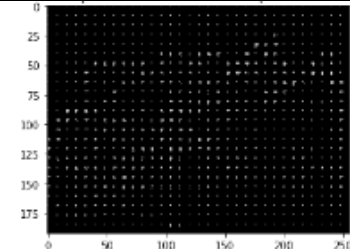
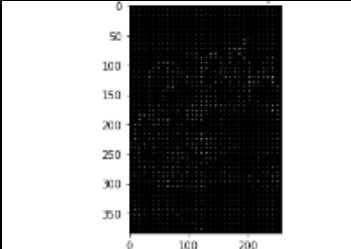
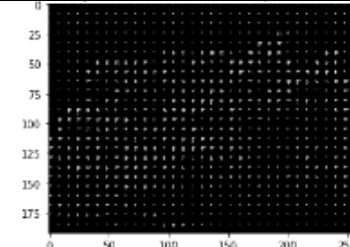
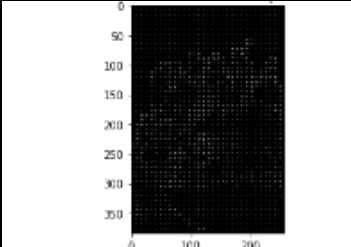
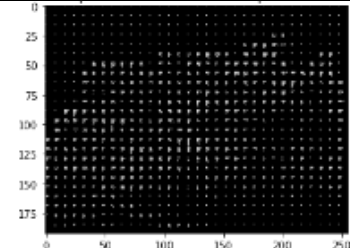
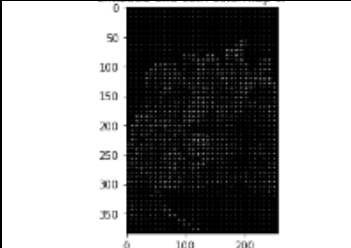
Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		
	25		
	50		
	75		

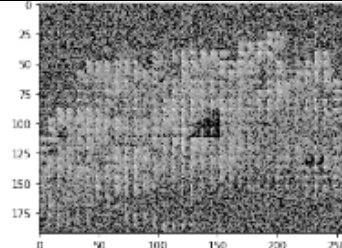
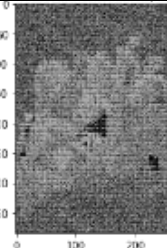
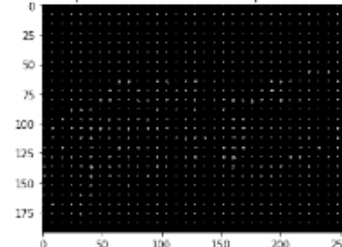
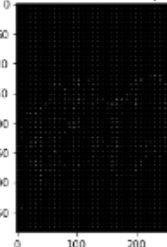
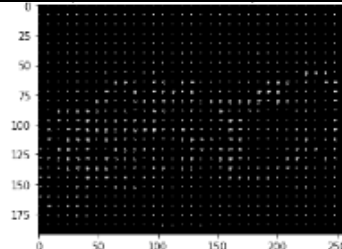
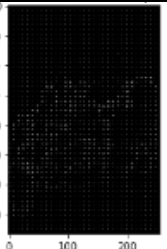
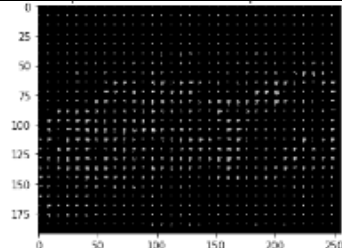
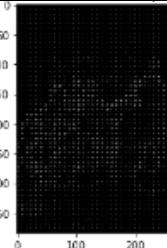
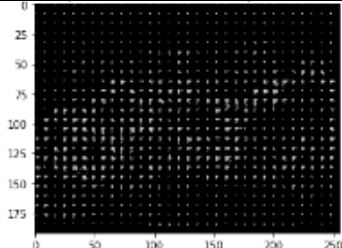
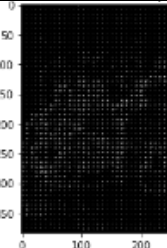
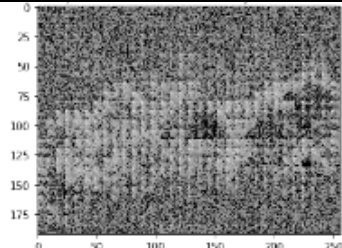
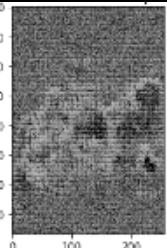
	100		
Cb	10		
	25		
	50		
	75		
	100		
Cr	10		

	25		
	50		
	75		
	100		

- peppers.bmp

Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		
	25		

	50		
	75		
	100		
Cb	10		
	25		
	50		
	75		

	100		
Cr	10		
	25		
	50		
	75		
	100		

Tal como podemos observar, à medida que o fator de qualidade diminui, a imagem fica progressivamente mais escura, o que indicia que sofre mais compressão destrutiva e como tal, qualidade mais baixa. Isto acontece, pois, a matriz de quantização nestes casos apresenta valores mais altos.

Entre as imagens resultantes de downsampling 4:2:2 e 4:2:0 não se veem diferenças no que diz respeito à quantização. A única diferença provém mesmo do downsampling: um tem mais linhas que o outro.

É notório também pela análise das imagens acima que a imagem resultante da aplicação da quantização é mais semelhante à original no canal Y. Isto acontece uma vez que, tal como já explicamos acima, este canal é o que contém a luminância, enquanto os Cb e Cr contêm a crominância. Como o olho humano é mais sensível à luminância que à crominância, comprime-se menos.

Pergunta 8.4 – Quantização vs DCT

Comparando com a pergunta 7 em que aplicamos DCT, percebemos que os resultados com fator de qualidade 100 são muito semelhantes (senão iguais) à aplicação da DCT em blocos 8x8. Isto acontece uma vez que aplicamos a matriz de quantização em blocos 8x8 e, sendo o fator de quantização 100, essa matriz é composta por uns. Ora, posto isto, a imagem resultante da aplicação dessa matriz de quantização será igual à imagem pré matriz de quantização.

Pergunta 9.3 - DPCM

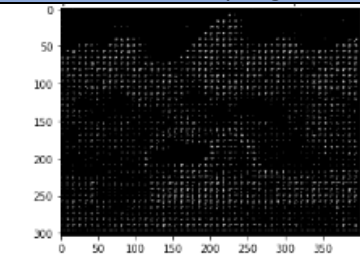
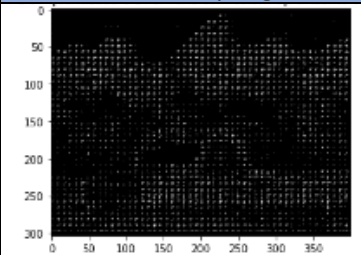
A DPCM consiste na codificação diferencial dos coeficientes DC consecutivos. Isto permite que haja uma gama de valores mais estreita e com menor variância, o que irá permitir uma maior compressibilidade devido à menor entropia entre os valores.

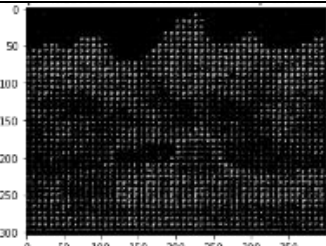
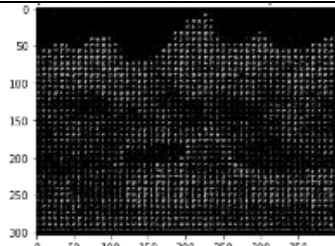
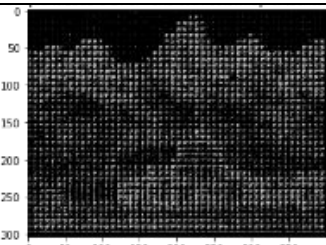
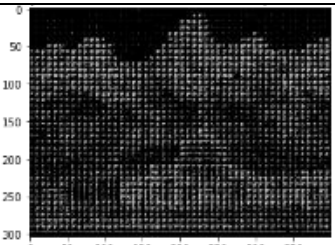
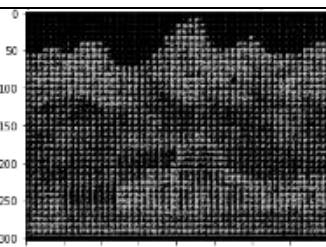
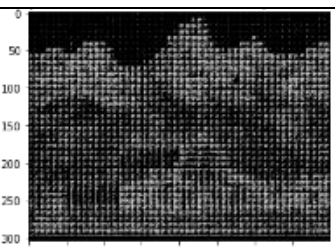
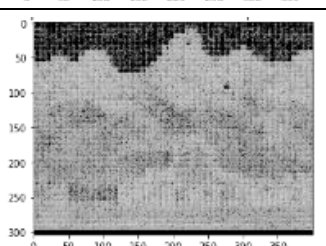
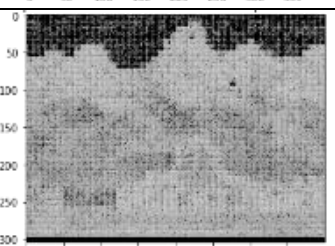

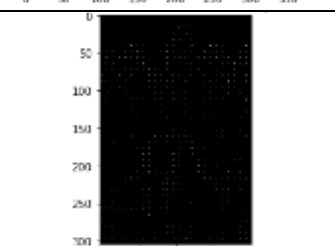
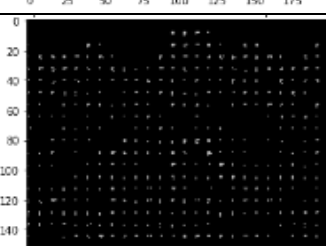
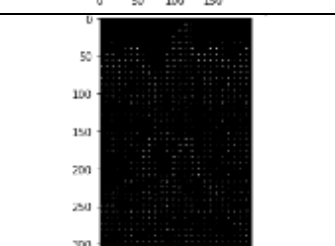
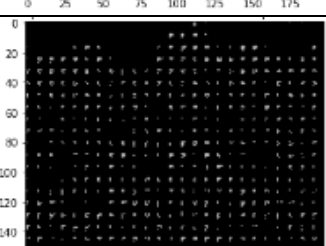
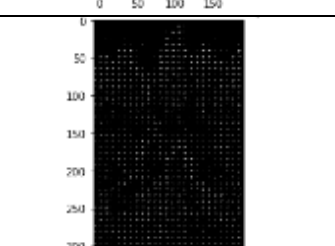
Dizemos que a matriz irá apresentar uma gama de valores mais estreita porque ao se fazer a diferença entre um coeficiente DC e o seu anterior, teremos uma menor gama de valores em relação à anterior onde ainda não tinha sido aplicada a codificação diferencial. Como consequência disto, será necessário um menor número de bits para representar a mesma.

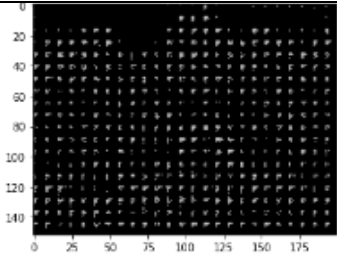
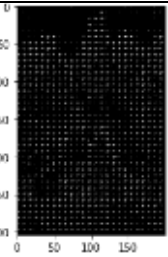
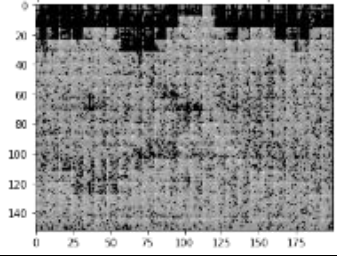
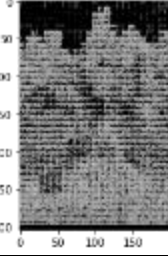
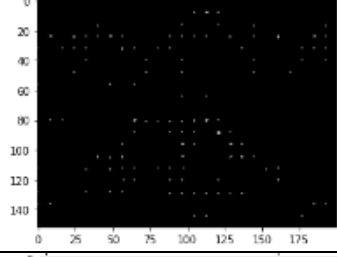
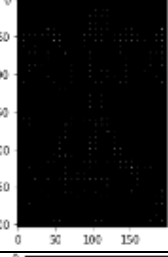
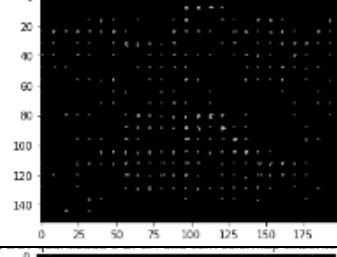
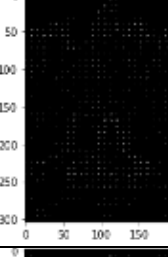
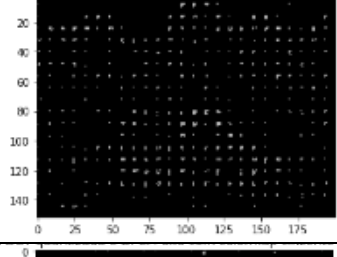
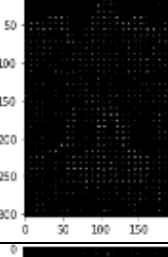
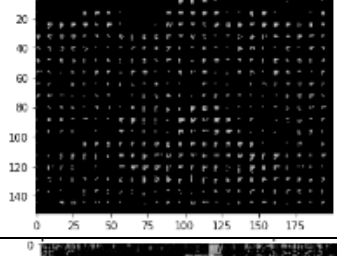
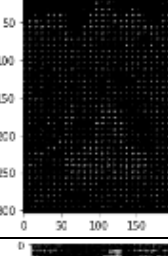
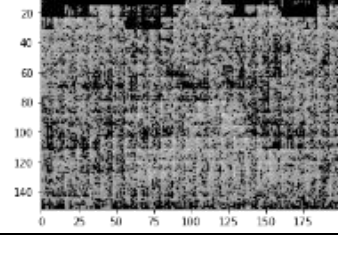
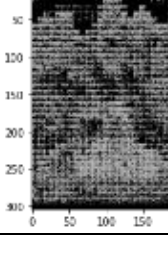
No caso de estarmos perante uma imagem suave, os valores dos coeficientes DC serão semelhantes entre si, o que após a codificação, serão gerados muitos valores próximos de 0, o que permite aplicar nestes o algoritmo de compressão RLE e no restante Huffman.

Os resultados obtidos foram os seguintes:

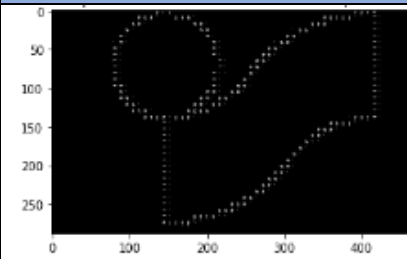
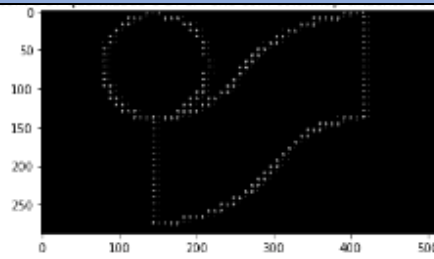
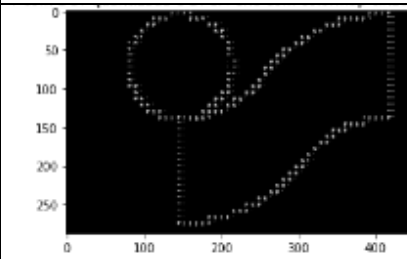
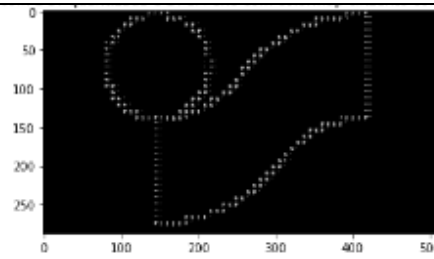
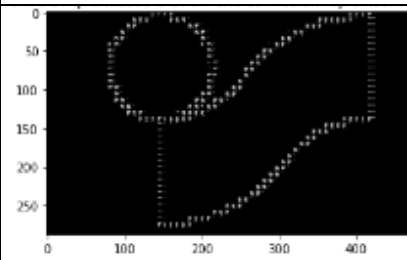
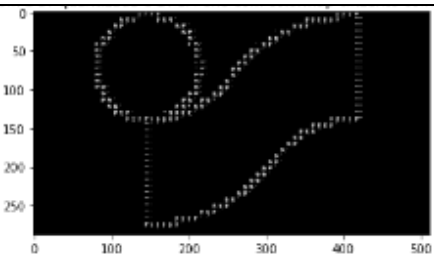
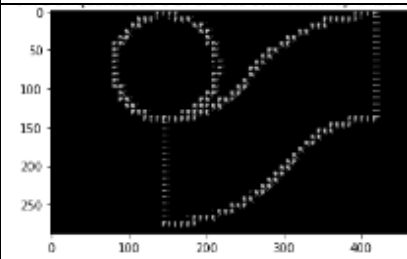
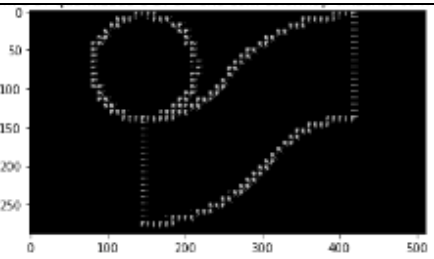
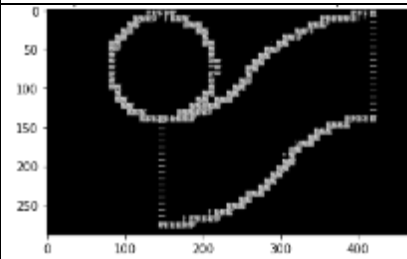
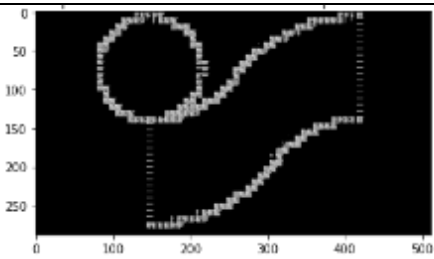
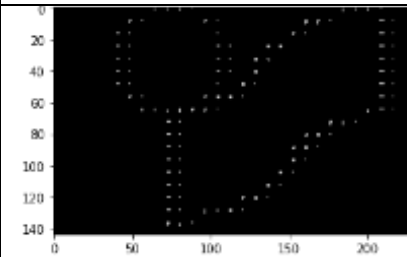
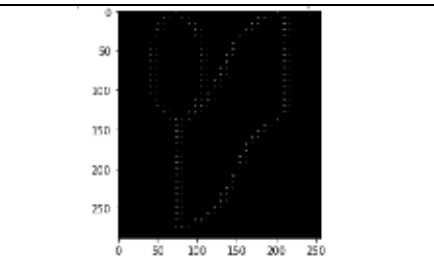
- barn_mountains.bmp

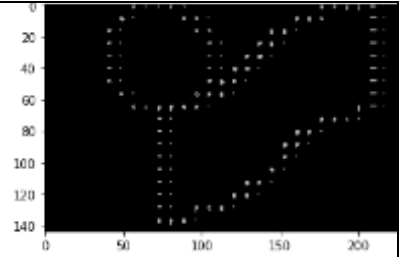
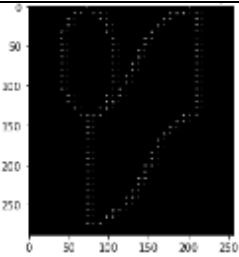
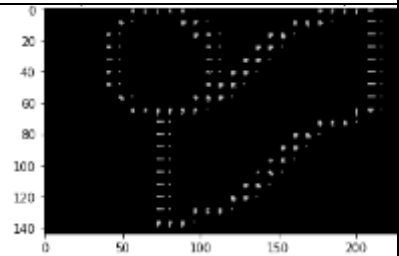
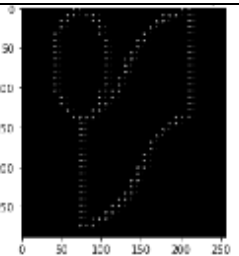
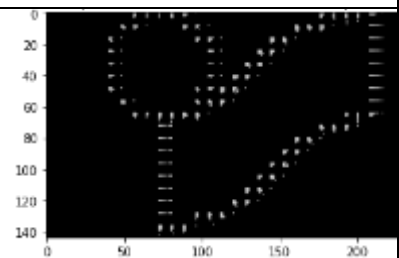
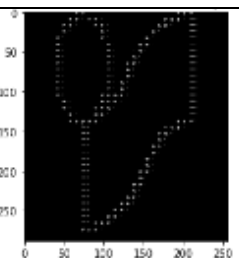
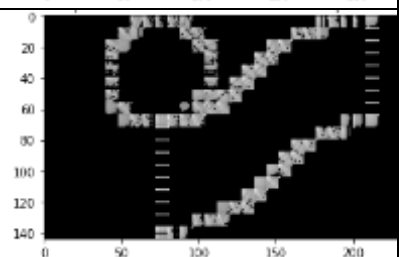
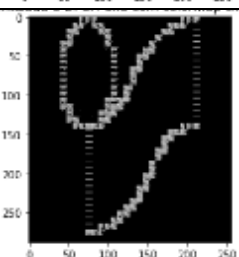
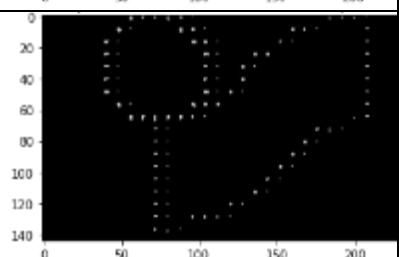
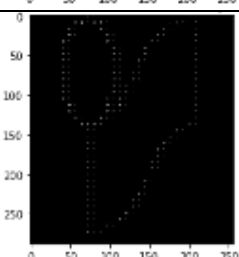
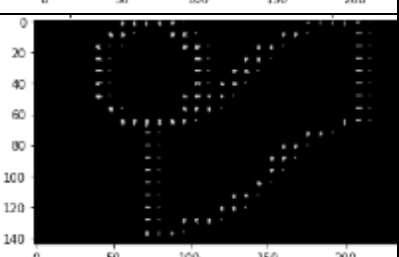
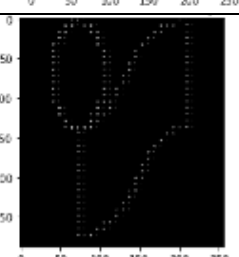
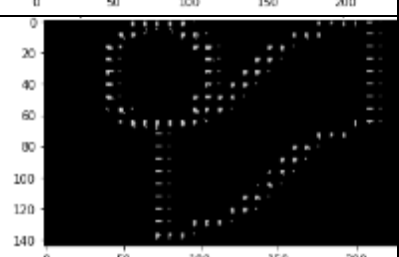

Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		

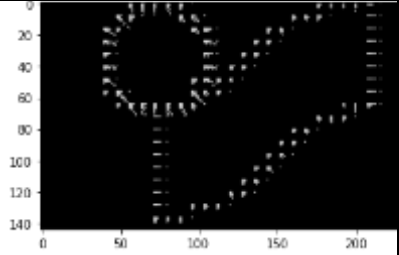
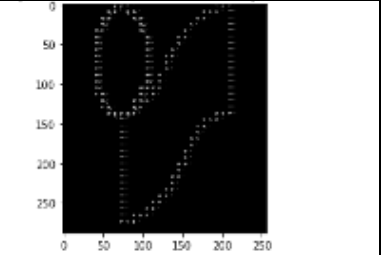
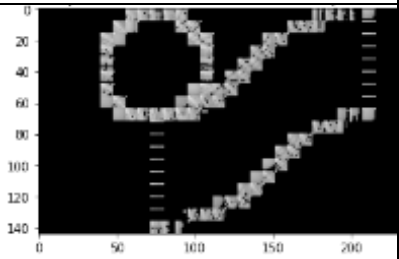
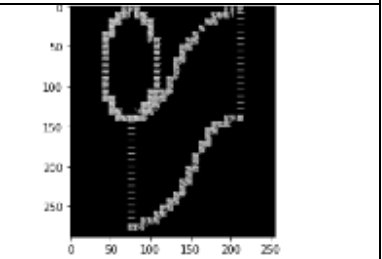
	25		
	50		
	75		
	100		
Cb	10		
	25		
	50		

	75		
	100		
Cr	10		
	25		
	50		
	75		
	100		

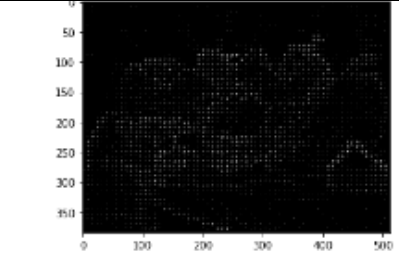
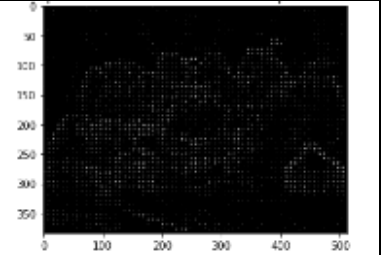
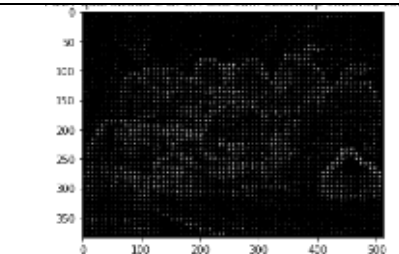
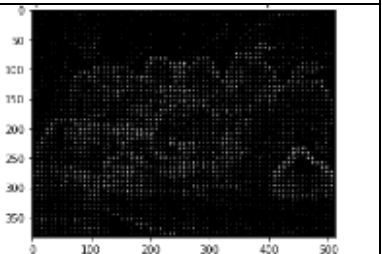
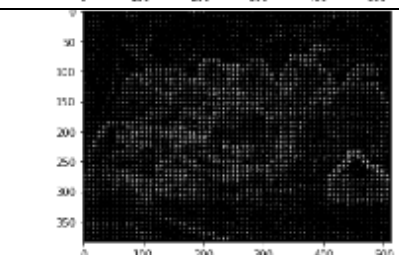
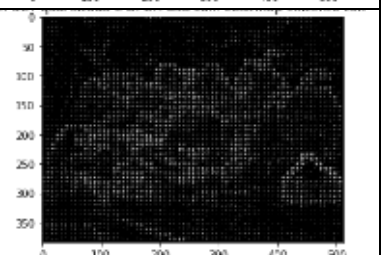
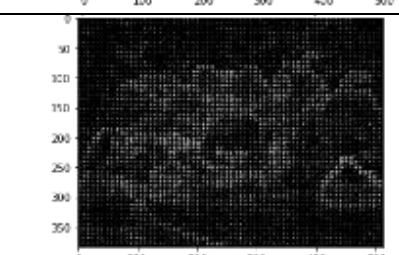
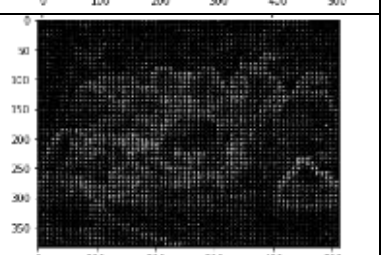
- logo.bmp

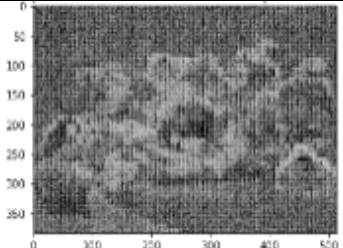
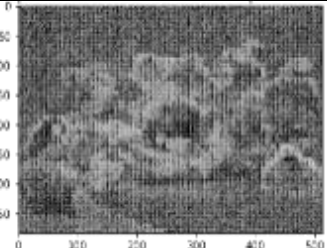
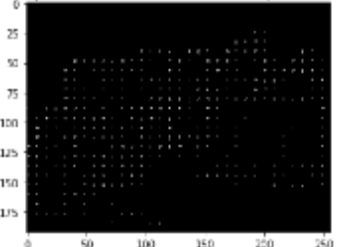
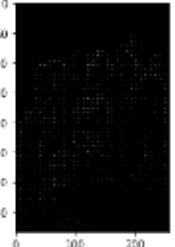
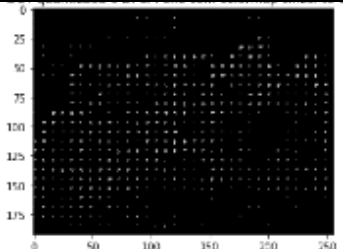
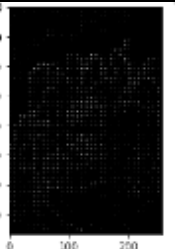
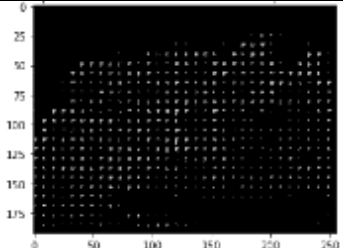

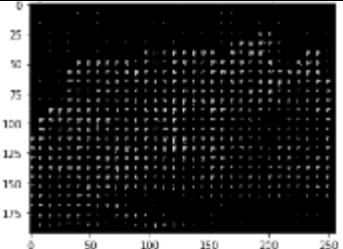
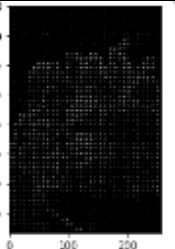
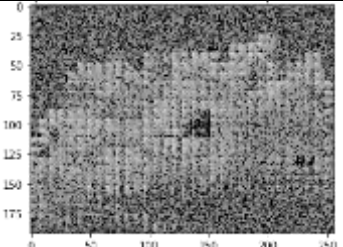

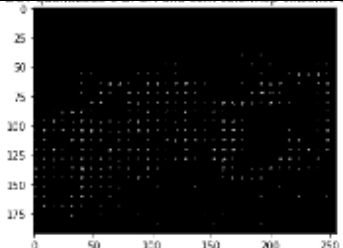

Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		
	25		
	50		
	75		
	100		
Cb	10		

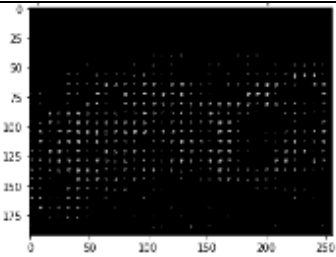
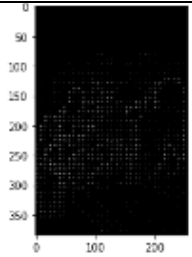
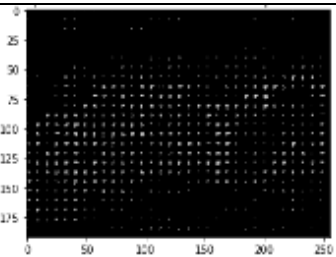
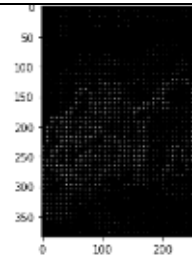
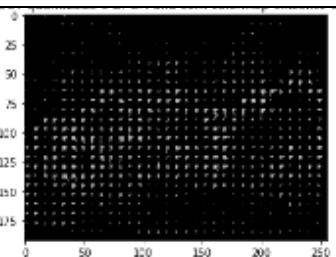
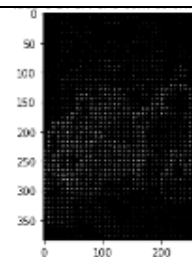
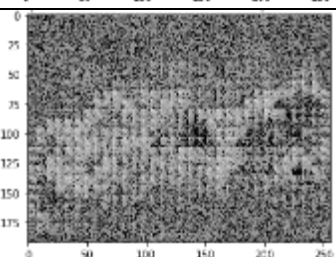
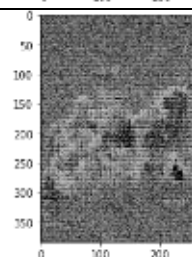
	25		
	50		
	75		
	100		
Cr	10		
	25		
	50		

	75		
	100		

- peppers.bmp

Canal	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
Y	10		
	25		
	50		
	75		

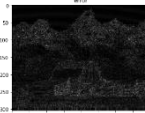
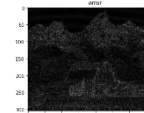
	100		
Cb	10		
	25		
	50		
	75		
	100		
Cr	10		

	25		
	50		
	75		
	100		

Comparando estas imagens com as que apresentamos na alínea anterior, verificamos que são muito parecidas. A única diferença recai no primeiro pixel de cada bloco. Essa pequena alteração ajuda imenso à compressão uma vez que permite aumentar a redundância, o que ajuda a compressão posterior através de mecanismos como RLE e códigos de Huffman.

Pergunta 10.2 – Erros e imagens descomprimidas

Obviamente, sendo o JPEG um algoritmo de compressão com perdas de informação, existem diferenças entre a imagem original e a imagem decodificada (erro). As seguintes imagens representam esse mesmo erro.

Imagem	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
barn_mountains	10		

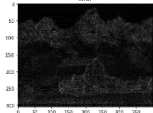
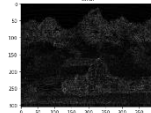
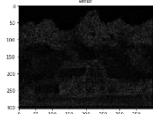
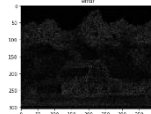
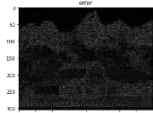
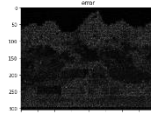
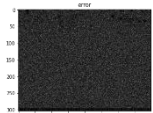
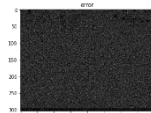
	25		
	50		
	75		
	100		

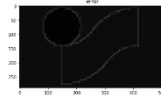
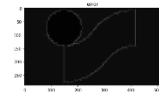
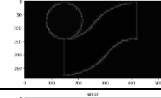
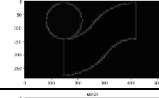
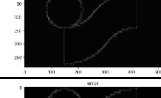
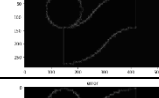
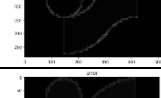

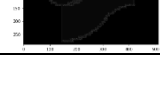
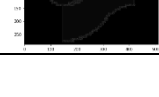
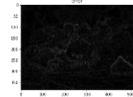
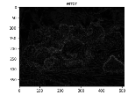
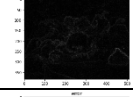
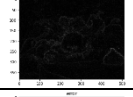
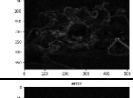
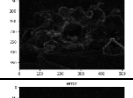
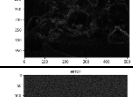
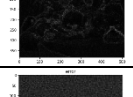
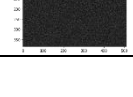
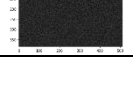
Imagem	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
logo	10		
	25		
	50		
	75		
	100		

Imagem	Fator de qualidade	Com downsampling 4:2:0	Com downsampling 4:2:2
peppers	10		
	25		
	50		
	75		
	100		

Uma vez que o matplotlib normaliza as matrizes, ele pega em qualquer valor que seja máximo nas matrizes e faz uma proporção com todos sendo esse máximo 255. Deste modo, mesmo que o valor seja quase nulo, vai estar representado a branco na imagem. Com isto, não conseguimos visualizar através das imagens acima a quantidade de erro existente.

Para conseguirmos entender, apresentamos de seguida os valores de um bloco da matriz de erro com downsampling 4:2:0 para a imagem barn_mountains.bmp:

Fator de qualidade	Com downsampling 4:2:0
10	[[3.02 3.02 3.02 3.02 3.031 3.031 5.031 5.031] [3.02 3.02 3.02 3.02 3.031 3.031 5.031 5.031] [2.319 4.02 5.607 5.607 5.031 5.031 4.031 4.031] [2.319 4.02 5.607 5.607 5.031 5.031 5.031 5.031] [5.031 5.031 5.031 5.031 5.607 5.607 5.607 5.607] [5.031 5.031 5.031 5.031 5.607 5.607 5.607 5.607] [5.031 5.031 6.618 6.618 5.607 5.607 5.607 5.607] [5.031 5.031 6.618 6.618 5.607 5.607 5.607 5.607]]
25	[[2.98 2.98 2.98 2.98 2.969 2.969 0.969 0.969] [2.98 2.98 2.98 2.98 2.969 2.969 0.969 0.969] [3.681 1.98 0.393 0.393 0.969 0.969 1.969 1.969] [3.681 1.98 0.393 0.393 0.969 0.969 0.969 0.969] [0.969 0.969 0.969 0.969 0.393 0.393 0.393 0.393] [0.969 0.969 0.969 0.969 0.393 0.393 0.393 0.393] [0.969 0.969 0.618 0.618 0.393 0.393 0.393 0.393] [0.969 0.969 0.618 0.618 0.393 0.393 0.393 0.393]]
50	[[1.10 1.10 1.10 1.10 1.11 1.11 3.11 3.11] [0.78 0.78 0.78 0.78 0.79 0.79 2.79 2.79] [0.50 1.1985 2.7855 2.7855 2.2095 2.2095 1.2095 1.2095] [1.267 0.43 2.02 2.02 1.44 1.44 1.44 1.44] [0.617 0.617 0.617 0.617 1.19 1.19 1.19 1.19] [0.1475 0.1475 0.1475 0.1475 0.428 0.428 0.428 0.428] [0.73 0.73 0.854 0.854 0.1568 0.1568 0.1568 0.1568] [1.0496 1.0496 0.537 0.537 0.47 0.47 0.47 0.47]]
75	[[0.9397 0.9397 0.9397 0.9397 0.9287 0.9287 1.071 1.071] [1.098 1.0980 1.0980 1.0980 1.087 1.087 0.91 0.91] [2.09 0.3907 1.196 1.196 0.620 0.620 0.3797 0.3797] [2.47 0.77 0.81 0.81 0.2379 0.2379 0.2379 0.2379] [0.1759 0.1759 0.1759 0.1759 0.400 0.400 0.400 0.400] [0.558 0.558 0.558 0.558 0.0177 0.0177 0.0177 0.0177] [0.85 0.85 0.736 0.736 0.27 0.27 0.27 0.27] [1.009 1.009 0.5777 0.5777 0.433 0.433 0.433 0.433]]
100	[[0.1769 0.0078 0.204 0.64 0.0168 0.147 0.286 0.261] [0.173 0.06 0.1 0.250 0.0347 0.673 0.203 0.249] [0.01 0.62 0.39 0.0647 0.02 0.449 0.273 0.440] [0.314 0.299 0.6067 0.469 0.24 0.413 0.238 0.5398] [0.10 0.198 0.117 0.330 0.0898 0.242 0.17 0.239] [0.171 0.0835 0.09 0.18 0.41 0.30 0.24 0.165] [0.09 0.21 0.07 0.161 0.22 0.27 0.08 0.08] [0.047 0.445 0.0347 0.16688 0.2688 0.156 0.239 0.1049]]

Analisando agora estes valores, fica evidente que as imagens apresentadas antes estavam com as suas matrizes normalizadas. É também notório que quanto menor o fator de qualidade, maior o erro, uma vez que existe mais compressão e consequente perda de informação.

Pode surgir a dúvida do porquê de com fator de qualidade 100 existir erro. Isso acontece, pois, apesar de a qualidade estar no máximo, existe downsampling que destrói sempre informação não recuperável.

Com isto, fomos ainda calcular vários tipos de erro:

- barn_mountains.bmp

4:2:0	Fator de Qualidade				
	10	25	50	75	100
MSE	745.1030	430.2006	293.2126	187.5073	55.5369
RMSE	27.2966	20.7413	17.1235	13.6933	7.4523
SNR	18.4641	20.8496	22.5144	24.4561	29.7405
PSNR	19.4086	21.7941	23.4590	25.4006	30.6850
4:2:2	Fator de Qualidade				
	10	25	50	75	100
MSE	712.3734	403.4538	267.0883	160.1619	24.3687
RMSE	26.6903	20.0861	16.3428	12.6555	4.9365
SNR	18.6592	21.1283	22.9197	25.1407	33.3179
PSNR	19.6037	22.0729	23.8643	26.0852	34.2625

- logo.bmp

4:2:0	Fator de Qualidade				
	10	25	50	75	100
MSE	207.8163	100.9359	80.9997	60.7448	43.0664
RMSE	14.4158	10.0467	8.99998	7.7939	6.5625
SNR	28.1553	31.2917	32.2473	33.4970	34.9907
PSNR	24.9540	28.0903	29.0460	30.2957	31.7894
4:2:2	Fator de Qualidade				
	10	25	50	75	100
MSE	166.5376	72.4833	56.4930	36.0339	18.4399
RMSE	12.9049	8.5137	7.5162	6.0028	4.2942
SNR	29.1170	32.7297	33.8122	35.7650	38.6745
PSNR	25.9157	29.5284	30.6109	32.5637	35.4732

- peppers.bmp

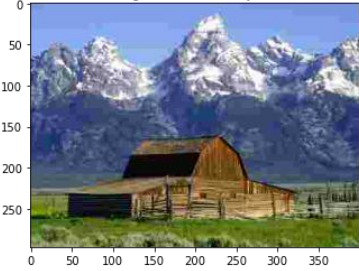
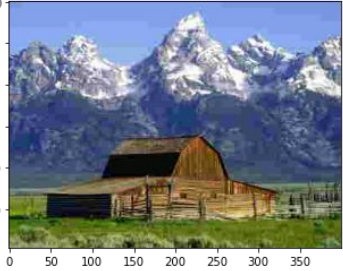
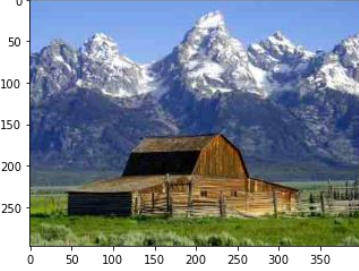
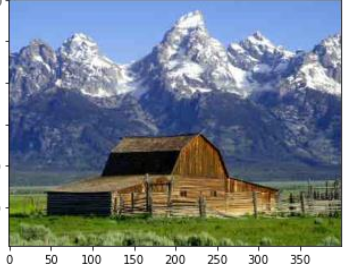


4:2:0	Fator de Qualidade				
	10	25	50	75	100
MSE	366.5626	188.4050	133.5154	104.5551	62.2776
RMSE	19.1458	13.7261	11.5549	10.2252	7.8916
SNR	19.2845	22.1750	23.6706	24.7325	26.9826
PSNR	22.4893	25.3799	26.87557	27.9374	30.1875
4:2:2	Fator de Qualidade				
	10	25	50	75	100

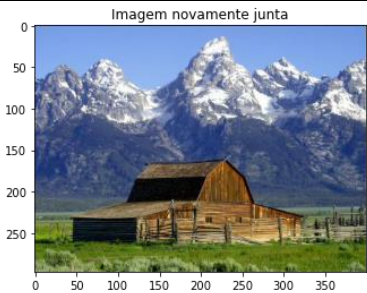
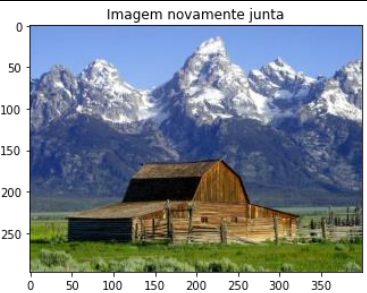
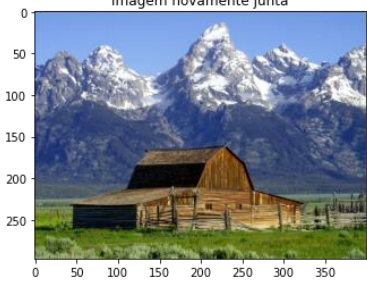
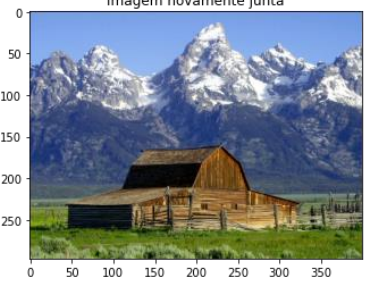
MSE	296.7947	140.4057	89.5071	64.2514	25.5789
RMSE	17.2277	11.8493	9.4608	8.0157	5.0576
SNR	20.2014	23.4521	25.4074	26.8471	30.8471
PSNR	23.4062	26.6570	28.6122	30.0520	34.0520

Tal como era espectável, para qualquer imagem, qualquer que seja o downsampling (4:2:2 ou 4:2:0), à medida que o fator de qualidade aumenta, o MSE e o RMSE diminuem e o SNR e o PSNR aumentam.

Estávamos também à espera que comparando os valores de erro entre downsampling 4:2:2 e 4:2:0, o que tem 4:2:0 apresenta valores de erro superiores uma vez que nessa fase não só é eliminada informação na direção horizontal como também vertical. Ao reconstruir, sendo que igualamos à coluna anterior, existirá sempre erro comparando com a imagem original.

De seguida, apresentamos os resultados da imagem barn_mountains.bmp descomprimida:

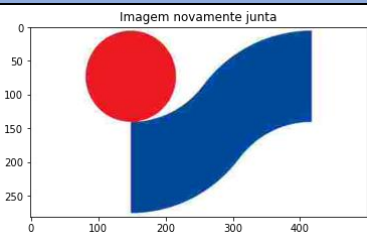
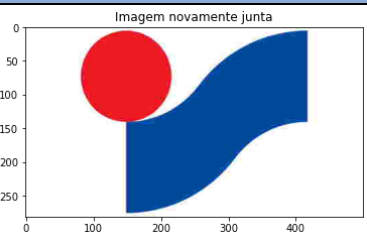
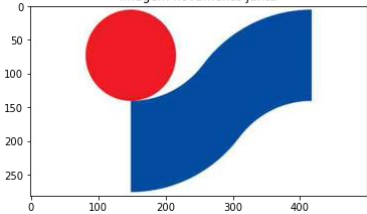
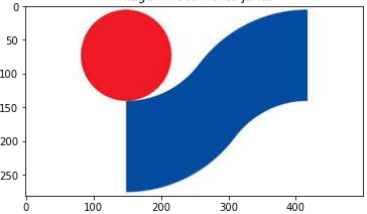
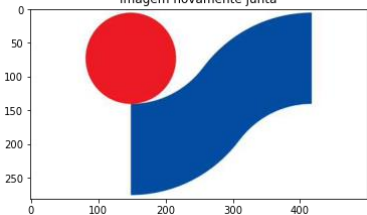
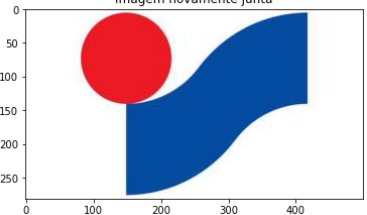
Fator de qualidade	Downsampling 4:2:0	Downsampling 4:2:2
10	<p>Imagem novamente junta</p> 	<p>Imagem novamente junta</p> 
25	<p>Imagem novamente junta</p> 	<p>Imagem novamente junta</p> 
50	<p>Imagem novamente junta</p> 	<p>Imagem novamente junta</p> 

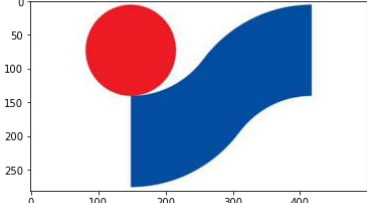
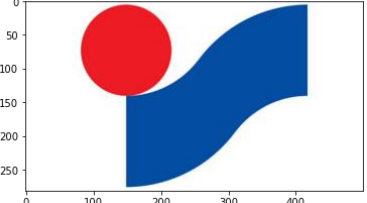


75		
100		

Tal como era espectável, quanto menor o fator de qualidade, menos nítida se torna a imagem e mais grão apresenta.

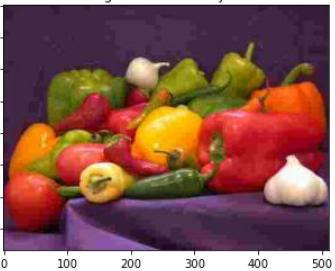





Se compararmos para o mesmo fator de qualidade, a imagem resultante da compressão com downsampling 4:2:0 com a de downsampling 4:2:2, percebemos que a de downsampling 4:2:0 é menos nítida. Esta situação também era espectável pois com esta proporção de downsampling perde-se mais informação que não se recupera.





Os resultados para o logo.bmp foram os seguintes:

Fator de qualidade	Downsampling 4:2:0	Downsampling 4:2:2
10		
25		
50		

75		
100		

Já os resultados para a imagem peppers.bmp são:

Fator de qualidade	Downsampling 4:2:0	Downsampling 4:2:2
10		
25		
50		

75	 <p>Imagem novamente junta</p>	 <p>Imagem novamente junta</p>
100	 <p>Imagem novamente junta</p>	 <p>Imagem novamente junta</p>

Nestas duas últimas tabelas percebemos que, tal como na barn_mountains.bmp, quanto menor o fator de qualidade, menos nítida e com mais grão se apresenta a imagem. Também no downsampling 4:2:0 a imagem perde mais qualidade.

Em suma, fomos percebendo ao longo deste trabalho prático que o codec JPEG é melhor em imagens suaves, uma vez que tem uma série de mecanismos que comprime de forma mais eficaz estes casos. Deste modo, não é de admirar que cheguemos ao fim e reparemos que a imagem que perdeu mais qualidade após descompressão é a barn_mountains.bmp uma vez que é a que apresenta transições mais abruptas.

Conclusão

Com este trabalho prático, ficamos mais cientes do funcionamento do *codec* de compressão de imagem JPEG e de como este é destrutivo. Através do nosso *dataset*, ficou bem notório que este codec funciona melhor com imagens suaves, uma vez que tira partido das transições pouco abruptas.

Deste modo, conseguimos entender melhor como funciona, o que nos permitiu entender de forma prática a matéria teórica lecionada nesta unidade curricular.

Referências

- “Lossy Data Compression: JPEG” in Stanford.
<https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossy/jpeg/coeff.htm> [24/03/2022]
- “RESOLUÇÃO ESPACIAL (DOWN SAMPLING E UP SAMPLING) NO PROCESSAMENTO DE IMAGEM” in Acervo Lima. <https://acervolima.com/resolucao-espacial-down-sampling-e-up-sampling-no-processamento-de-imagem/> [24/03/2022]
- “Transformada discreta de cosseno” in *pt* - Wikipedia.
https://pt.wikipedia.org/wiki/Transformada_discreta_de_cosseno#Compress%C3%A3o [24/03/2022]
- Material fornecido pelos docentes da cadeira